Project Planning Phase

Technology Stack (Architecture & Stack)

Date: 4 June 2025
Project Name: A Customizable Snack Ordering and Delivery App
Maximum Marks: 4 Marks

Technical Architecture

The proposed solution is a snack ordering Android application designed to streamline the process of browsing, customizing, and ordering snacks for events or casual moments. It consists of a layered architecture with the following modules: User Interface (Android), Business Logic (Kotlin), Data Management (in-memory & SharedPreferences), and optional Firebase backend integration. It ensures real-time cart updates, secure login, and delivery tracking. The design supports modular extension for features like payment gateways, order history, and user feedback.

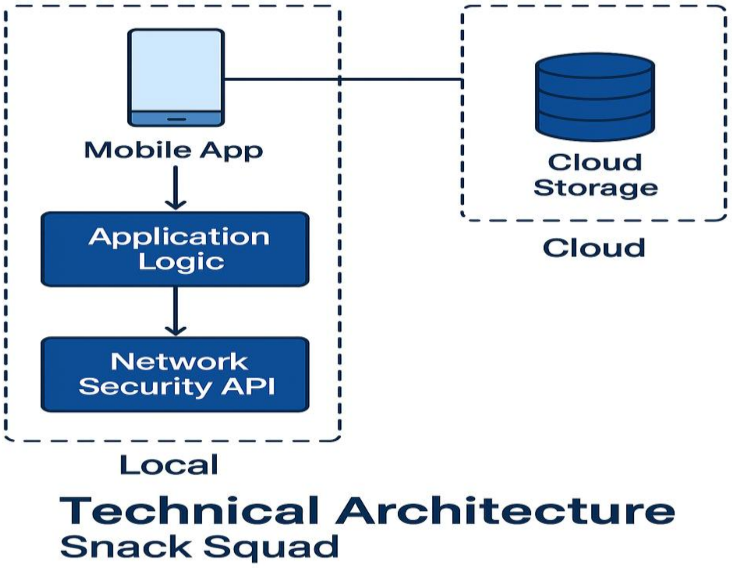

Table-1: Components & Technologies

S.NO	Component	Description	Technology
1	User Interface	Native Android app UI using XML layouts	XML, Android View System
2	Application Logic-1	Snack listing, filtering, cart handling	Kotlin, RecyclerView
3	Application Logic-2	User login/register, validation, session persistence	Kotlin, SharedPreferences
4	Application Logic-3	Payment validation, multiple payment methods	Kotlin, Material Components
5	Database	Temporary cart and user data storage	SharedPreferences (local storage)
6	Cloud Database	Optional – For real-time user/order sync	Firebase (future integration)
7	File Storage	App icon, snack images, assets	Android Res/Drawable, Assets
8	External API-1	Not used currently (static snack data used)	N/A
9	External API-2	Not used	N/A
10	Machine Learning Model	Snack image rendering	Glide (optional), ImageView
11	Infrastructure	Android SDK and Gradle-based build system	Android Studio, Gradle, Emulator

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology / Approach
1	Open-Source Frameworks	Libraries and dependencies	AndroidX, Material Components, Kotlin Std
2	Security Implementations	Input validation, safe preferences	Regex Validation, Android Context Secure
3	Scalable Architecture	Modular activity-based system (can scale to Firebase backend)	MVVM (future-ready), Clean Activity Design
4	Availability	App can work offline for browsing and session persistence	SharedPreferences, Local Cache
5	Performance	Smooth UI interaction and fast navigation	RecyclerView, Lazy Loading (Images)

References:

- https://developer.android.com/guide
- https://material.io/components
- https://firebase.google.com/docs
- https://developer.android.com/studio